

Chương 1

Ôn Tập

0. BÀI TẬP

- ♦ Bài 1: Viết chương trình nhập vào một phân số. Hãy cho biết phân số đó là phân số âm hay dương hay bằng không.
- ♦ Bài 2: Viết chương trình nhập vào một ngày. Tìm ngày kế tiếp và xuất kết quả.
- ♦ Bài 3: Viết chương trình nhập tọa độ hai điểm trong không gian. Tính khoảng cách giữa chúng và xuất kết quả.

0. BÀI TẬP

- ♦ Bài 4: Viết chương trình nhập vào một ngày. Tìm ngày hôm qua và xuất kết quả.
- ♦ Bài 5: Viết chương trình nhập vào 2 phân số. Tìm phân số lớn nhất và kết quả.
- ♦ Bài 6: Viết chương trình nhập vào 2 số phức. Tính tổng, hiệu, tích và xuất kết quả.
- ♦ Bài 7: Viết chương trình nhập tọa độ 3 đỉnh A,B,C của 1 tam giác trong mặt phẳng Oxy. Tính chu vi, diện tích và tìm tọa độ trọng tâm.

0. BÀI TẬP

- ♦ Bài 8: Viết chương trình nhập tọa tâm và bán kính của một đường tròn. Tính diện tích và chu vi của đường tròn.
- ♦ Bài 9: Viết chương trình nhập vào hai phân số. Tính tổng, hiệu, tích, thương giữa chúng và xuất kết quả.

HƯỚNG DẪN BÀI 1

- ♦ **Bài 1: Viết chương trình nhập vào một phân số. Hãy cho biết phân số đó là phân số âm hay dương hay bằng không.**

```
1. #include <stdio.h>
2. #include <conio.h>
3. struct phanso
4. {
5.     int tu;
6.     int mau;
7. };
8. typedef struct phanso PHANSO;
```

HƯỚNG DẪN BÀI 1

1. `void Nhap (PHANSO&) ;`
2. `void Xuat (PHANSO) ;`
3. `int XetDau (PHANSO) ;`

HƯỚNG DẪN BÀI 1

```
11. void main()  
12. {  
13.     PHANSO a;  
14.     Nhap(a);  
15.     int kq = XetDau(a);  
16.     switch(kq)  
17.     {  
18.         case 1: printf("");  
19.             break;  
20.         case -1: printf("");  
21.             break;  
22.         case 0: printf("");  
23.             break;  
24.     }  
25. }
```

HƯỚNG DẪN BÀI 1

♦ Định nghĩa hàm nhập phân số

```
1. void Nhap(PHANSO &x)
2. {
3.     printf("Nhap tu: ");
4.     scanf("%d",&x.tu);
5.     printf("Nhap mau: ");
6.     scanf("%d",&x.mau);
7. }
```


HƯỚNG DẪN BÀI 1

♦ Định nghĩa hàm xuất phân số

```
1. void Xuat(PHANSO x)
2. {
3.     printf("\n Tu: %d", x.tu);
4.     printf("\n Mau: %d", x.mau);
5. }
```

HƯỚNG DẪN BÀI 1

```
1. int XetDau(PHANSO x)
2. {
3.     if (x.tu*x.mau>0)
4.         return 1;
5.     if (x.tu*x.mau<0)
6.         return -1;
7.     return 0;
8. }
```

HƯỚNG DẪN BÀI 2

♦ Bài 2: Viết chương trình nhập vào một ngày. Tìm ngày kế tiếp và xuất kết quả.

```
1. #include <stdio.h>
2. #include <conio.h>
3. struct ngay
4. {
5.     int ng;
6.     int th;
7.     int nm;
8. };
9. typedef struct ngay NGAY;
```

HƯỚNG DẪN BÀI 2

1. void Nhap (NGAY&) ;
2. void Xuat (NGAY) ;
3. NGAY KeTiep (NGAY) ;
4. int ktNhuan (NGAY) ;
5. int sttTrongNam (NGAY) ;
6. long SoThuTu (NGAY) ;
7. NGAY TimNgay (int, int) ;
8. NGAY TimNgay (long) ;

HƯỚNG DẪN BÀI 2

```
1. void main()  
2. {  
3.     NGAY a;  
4.     Nhap(a);  
5.     NGAY kq = KeTiep(a);  
6.     printf("Ngay ban dau:");  
7.     Xuat(a);  
8.     printf("\nNgay ke tiep:");  
9.     Xuat(kq);  
10. }
```

HƯỚNG DẪN BÀI 2

♦ Định nghĩa hàm

```
1. void Nhap (NGAY &x)
2. {
3.     printf("Nhap ngay: ");
4.     scanf ("%d", &x.ng);
5.     printf("Nhap thang: ");
6.     scanf ("%d", &x.th);
7.     printf("Nhap nam: ");
8.     scanf ("%d", &x.nm);
9. }
```

HƯỚNG DẪN BÀI 2

♦ Định nghĩa hàm

```
1. void Xuat (NGAY x)
2. {
3.     printf ("\nNgày: %d", x.ng) ;
4.     printf ("\nThang: %d", x.th) ;
5.     printf ("\nNam: %d", x.nm) ;
6. }
```

HƯỚNG DẪN BÀI 2

- ♦ Một năm được gọi là năm nhuận khi thoả một trong hai điều kiện.
 - Điều kiện 1: Năm chia hết cho 4 và không chia hết cho 100.
 - Điều kiện 2: Năm chia hết cho 400.
- ♦ Ví dụ 1: 1996 nhuận (điều kiện 1)
- ♦ Ví dụ 2: 2000 nhuận (điều kiện 2)
- ♦ Ví dụ 3: 1900 ko nhuận.
- ♦ Định nghĩa hàm

```
1. int ktnhuan (NGAY x)
2. {
3.     if (x.nm%4==0 &&
           x.nm%100!=0)
4.         return 1;
5.     if (x.nm%400==0)
6.         return 1;
7.     return 0;
8. }
```



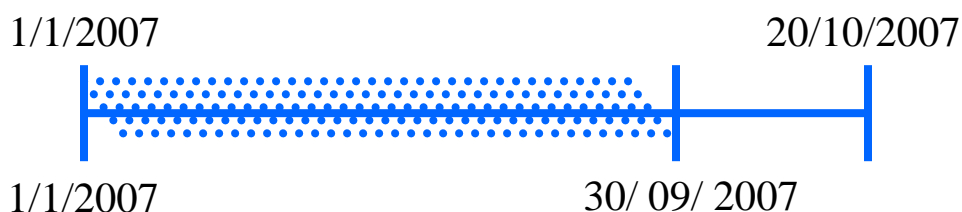
```
11. NGAY KeTiep (NGAY x)
12. {
13.     int NgayThang[12]={31,28,31,
        30,31,30,31,31,30,31,30,31};
14.     if (ktNhuan(x)==1)
15.         NgayThang[1] = 29;
16.     x.ng++;
17.     if (x.ng>NgayThang[x.th-1])
18.     {
19.         x.th++;
20.         if (x.th>12)
21.         {
22.             x.nm++;
23.             x.th = 1;
24.         }
25.         x.ng = 1;
26.     }
27.     return x;
28. }
```

```
11. NGAY HomQua (NGAY x)
12. {
13.     int NgayThang[12]={31,28,31,
        30,31,30,31,31,30,31,30,31};
14.     if (ktNhuan(x)==1)
15.         NgayThang[1] = 29;
16.     x.ng--;
17.     if (x.ng==0)
18.     {
19.         x.th--;
20.         if (x.th==0)
21.         {
22.             x.nm--;
23.             x.th = 12;
24.         }
25.         x.ng = NgayThang[x.th-1];
26.     }
27.     return x;
28. }
```

HƯỚNG DẪN BÀI 2

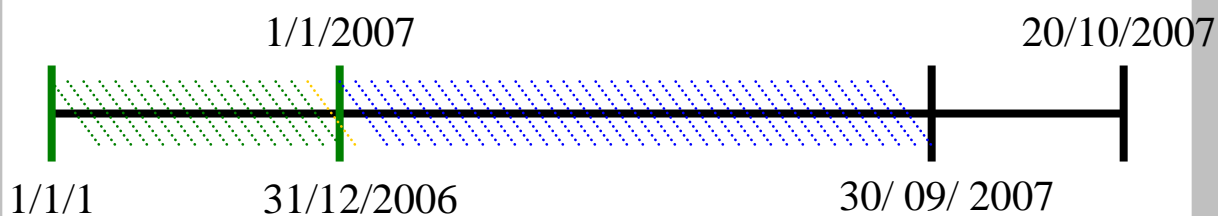
♦ Định nghĩa hàm

```
1. int sttTrongNam (NGAY x)
2. {
3.     int ngaythang[12]={31,28,31,
4.         30,31,30,31,31,30,31,30,31};
5.     if (ktNhuan(x)==1)
6.         ngaythang[1]=29;
7.     int stt = 0;
8.     for(int i=1;i<=x.th-1;i++)
9.         stt=stt+ngaythang[i-1];
10.    return (stt + x.ng);
11. }
```



HƯỚNG DẪN BÀI 2

```
1. long SoThuTu (NGAY x)
2. {
3.     long stt = 0;
4.     for (int i=1; i<=x.nm-1; i++)
5.     {
6.         stt = stt + 365;
7.         NGAY temp = {1,1,i};
8.         if (ktnhuan (temp) == 1)
9.             stt = stt+1;
10.    }
11.    return (stt+sttTrongNam (x) );
12. }
```



HƯỚNG DẪN BÀI 2

♦ Định nghĩa hàm

```
1. NGAY TimNgay(int nam,int stt)
2. {
3.     int ngaythang[12]={31,28,31,
4.         30,31,30,31,31,30,31,30,31};
5.     NGAY temp = {1,1,nam};
6.     if(ktNhuan(temp)==1)
7.         ngaythang[1] = 29;
8.     temp.th=1;
9.     while(stt-ngaythang[temp.th-1]>0)
10.    {
11.        stt=stt-ngaythang[temp.th-1];
12.        temp.th++;
13.    }
14.    temp.ng = stt;
15.    return temp;
```

HƯỚNG DẪN BÀI 2

♦ Định nghĩa hàm

```
1.  NGAY TimNgay(long stt)
2.  {
3.      int nam = 1;
4.      int sn = 365;
5.      while(stt-sn>0)
6.      {
7.          stt = stt - sn;
8.          nam++;
9.          sn = 365;
10.         NGAY temp = {1,1,nam};
11.         if( ktnhuan(temp)==1)
12.             sn = 366;
13.     }
14.     return TimNgay(nam, (int)stt);
15. }
```

HƯỚNG DẪN BÀI 2

♦ Định nghĩa hàm

```
1. NGAY KeTiep (NGAY x)
2. {
3.     long stt=SoThuTu(x);
4.     stt=stt+1;
5.     return TimNgay(stt);
6. }
```

HƯỚNG DẪN BÀI 3

- ♦ **Bài 3: Viết chương trình nhập tọa độ hai điểm trong không gian. Tính khoảng cách giữa chúng và xuất kết quả.**

```
1. #include <stdio.h>
2. #include <conio.h>
3. struct diemkg
4. {
5.     float x;
6.     float y;
7.     float z;
8. };
9. typedef struct diemkg DIEMKG;
```


HƯỚNG DẪN BÀI 3

1. `void Nhap(DIEMKG&) ;`
2. `void Xuat(DIEMKG) ;`
3. `float KhoangCach(DIEMKG,
DIEMKG) ;`

HƯỚNG DẪN BÀI 3

```
1. void main()  
2. {  
3.     DIEMKG A, B;  
4.     Nhap (A) ;  
5.     Nhap (B) ;  
6.     float kq=KhoangCach (A, B) ;  
7.     Xuat (A) ;  
8.     Xuat (B) ;  
9.     printf ("\n...: %f", kq) ;  
10. }
```

HƯỚNG DẪN BÀI 3

◆ Định nghĩa hàm

```
1. void Nhap (DIEMKG &P)
2. {
3.     float temp;
4.     printf("Nhap x: ");
5.     scanf("%f", &temp);
6.     P.x = temp;
7.     printf("Nhap y: ");
8.     scanf("%f", &temp);
9.     P.y = temp;
10.    printf("Nhap z: ");
11.    scanf("%f", &temp);
12.    P.z = temp;
13. }
```

HƯỚNG DẪN BÀI 3

♦ Định nghĩa hàm

```
1. void Xuat(DIEMKG P)
2. {
3.     printf("\n x=%f", P.x);
4.     printf("\n y=%f", P.y);
5.     printf("\n z=%f", P.z);
6. }
```

HƯỚNG DẪN BÀI 3

♦ Định nghĩa hàm

```
1. float KhoangCach(DIEMKG P, DIEMKG Q)
2. {
3.     return sqrt((Q.x-P.x)*(Q.x-P.x)
4.                 + (Q.y-P.y)*(Q.y-P.y)
5.                 + (Q.z-P.z)*(Q.z-P.z));
6. }
```

HƯỚNG DẪN BÀI 4

♦ Bài 4: Viết chương trình nhập vào một ngày. Tìm ngày hôm qua và xuất kết quả.

```
1. #include <stdio.h>
2. #include <conio.h>
3. struct ngay
4. {
5.     int ng;
6.     int th;
7.     int nm;
8. };
9. typedef struct ngay NGAY;
```

HƯỚNG DẪN BÀI 4

1. void Nhap (NGAY&) ;
2. void Xuat (NGAY) ;
3. NGAY TruocDo (NGAY) ;
4. int sttTrongNam (NGAY) ;
5. long SoThuTu (NGAY) ;
6. NGAY TimNgay (int, int) ;
7. NGAY TimNgay (long) ;

HƯỚNG DẪN BÀI 4

```
1. void main()  
2. {  
3.     NGAY a;  
4.     Nhap(a);  
5.     NGAY kq = TruocDo(a);  
6.     printf("Ngay ban dau:");  
7.     Xuat(a);  
8.     printf("\nNgay ke tiep:");  
9.     Xuat(kq);  
10. }
```


HƯỚNG DẪN BÀI 5

- ♦ **Bài 5: Viết chương trình nhập vào 2 phân số. Tìm phân số lớn nhất và kết quả.**

```
1. #include <stdio.h>
2. #include <conio.h>
3. struct phanso
4. {
5.     int tu;
6.     int mau;
7. };
8. typedef struct phanso PHANSO;
```

HƯỚNG DẪN BÀI 5

1. `void Nhap (PHANSO&) ;`
2. `void Xuat (PHANSO) ;`
3. `int SoSanh (PHANSO, PHANSO) ;`

HƯỚNG DẪN BÀI 5

```
1. void main()  
2. {  
3.     PHANSO a,b;  
4.     printf("\n Nhap...:");  
5.     Nhap(a);  
6.     printf("\n Nhap...:");  
7.     Nhap(b);  
8.     int kq = SoSanh(a,b);  
9.     printf("\nPhan so...:");  
10.    if(kq>=0)  
11.        Xuat(a);  
12.    else  
13.        Xuat(b);  
14. }
```

HƯỚNG DẪN BÀI 5

♦ Định nghĩa hàm nhập phân số

```
1. void Nhap (PHANSO &x)
2. {
3.     printf("Nhap tu: ");
4.     scanf("%d", &x.tu);
5.     printf("Nhap mau: ");
6.     scanf("%d", &x.mau);
7. }
```

HƯỚNG DẪN BÀI 5

♦ Định nghĩa hàm xuất phân số

```
1. void Xuat(PHANSO x)
2. {
3.     printf("\n Tu: %d", x.tu);
4.     printf("\n Mau: %d", x.mau);
5. }
```

HƯỚNG DẪN BÀI 5

```
11. int SoSanh (PHANSO x, PHANSO y)
12. {
13.     float a=(float)x.tu/x.mau;
14.     float b=(float)y.tu/y.mau;
15.     if (a>b)
16.         return 1;
17.     if (a<b)
18.         return -1;
19.     return 0;
20. }
```

HƯỚNG DẪN BÀI 6

- ♦ **Bài 6: Viết chương trình nhập vào 2 số phức. Tính tổng, hiệu, tích và xuất kết quả.**

```
1. #include <stdio.h>
2. #include <conio.h>
3. struct sophuc
4. {
5.     float thuc;
6.     float ao;
7. };
8. typedef struct sophuc SOPHUC;
```

HƯỚNG DẪN BÀI 6

1. void Nhap (SOPHUC&) ;
2. void Xuat (SOPHUC) ;
3. SOPHUC Tong (SOPHUC, SOPHUC) ;
4. SOPHUC Hieu (SOPHUC, SOPHUC) ;
5. SOPHUC Tich (SOPHUC, SOPHUC) ;

HƯỚNG DẪN BÀI 6

```
1. void main()  
2. {  
3.     SOPHUC a,b,kq;  
4.     printf("\n Nhap...:");  
5.     Nhap(a);  
6.     printf("\n Nhap...:");  
7.     Nhap(b);  
8.     kq = Tong(a,b);  
9.     printf("\n Tong...:");  
10.    Xuat(kq);  
11.    kq = Hieu(a,b);  
12.    printf("\n Hieu...:");  
13.    Xuat(kq);  
14.    kq = Tich(a,b);  
15.    printf("\n Tich...:");  
16.    Xuat(kq);  
17. }
```

HƯỚNG DẪN BÀI 6

♦ Định nghĩa hàm

```
1. void Nhap (SOPHUC &x)
2. {
3.     float temp;
4.     printf("Nhap thuc: ");
5.     scanf("%f",&temp);
6.     x.thuc = temp;
7.     printf("Nhap ao: ");
8.     scanf("%f",&temp);
9.     x.ao = temp;
10. }
```

HƯỚNG DẪN BÀI 6

♦ Định nghĩa hàm

```
1. void Xuat(SOPHUC x)
2. {
3.     printf("\nThuc=%f", x.thuc);
4.     printf("\nAo = %f", x.ao);
5. }
```

HƯỚNG DẪN BÀI 6

- ♦ Ví dụ: $3+5i$ ^a
- ♦ $7+10i$ ^b
- ♦ Tổng: $10+15i$ temp

♦ Định nghĩa hàm

```
1. SOPHUC Tong (SOPHUC a, SOPHUC b)
2. {
3.     SOPHUC temp;
4.     temp.thuc = a.thuc + b.thuc;
5.     temp.ao = a.ao + b.ao;
6.     return temp;
7. }
```

HƯỚNG DẪN BÀI 6

♦ Ví dụ: $3 + 5i$

♦
$$\frac{7 + 10i}{-4 - 5i}$$

♦ Hiệu: $-4 - 5i$

♦ Định nghĩa hàm

```
1. SOPHUC Hieu (SOPHUC a, SOPHUC b)
2. {
3.     SOPHUC temp;
4.     temp.thuc = a.thuc - b.thuc;
5.     temp.ao = a.ao - b.ao;
6.     return temp;
7. }
```

HƯỚNG DẪN BÀI 6

- ♦ Ví dụ: $a + bi$
- ♦
$$\frac{c + di}{(a+bi)(c+di)}$$
- ♦ Tích $(a+bi)(c+di)$
- ♦ $a(c+di)+bi(c+di)$
- ♦ $ac+adi+bci+bdi^2$
- ♦ $(ac-bd)+(ad+bc)i$

♦ Định nghĩa hàm

```

1. SOPHUC Tich (SOPHUC x, SOPHUC y)
2. {
3.     SOPHUC temp;
4.     temp.thuc=x.thuc*y.thuc-x.ao*y.ao;
5.     temp.ao=x.thuc*y.ao+x.ao*y.thuc;
6.     return temp;
7. }
```

HƯỚNG DẪN BÀI 7

- ♦ **Bài 7: Viết chương trình nhập tọa độ 3 đỉnh A,B,C của 1 tam giác trong mặt phẳng Oxy. Tính chu vi, diện tích và tìm tọa độ trọng tâm.**

```
1. #include <stdio.h>
2. #include <conio.h>
3. struct diem
4. {
5.     float x;
6.     float y;
7. };
8. typedef struct diem DIEM;
```

HƯỚNG DẪN BÀI 7

```
1. struct tamgiac
2. {
3.     DIEM A;
4.     DIEM B;
5.     DIEM C;
6. };
7. typedef struct tamgiac TAMGIAC;
8. void Nhap(DIEM&) ;
9. void Xuat(DIEM) ;
10. float KhoangCach(DIEM, DIEM) ;
11. void Nhap(TAMGIAC&) ;
12. void Xuat(TAMGIAC) ;
13. float ChuVi(TAMGIAC) ;
14. float DienTich(TAMGIAC) ;
15. DIEM TrongTam(TAMGIAC) ;
```

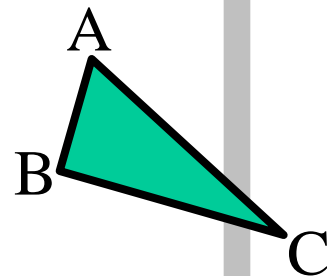

HƯỚNG DẪN BÀI 7

```
1. void main()  
2. {  
3.     TAMGIAC tg;  
4.     Nhap(tg);  
5.     Xuat(tg);  
6.     float cv = ChuVi(tg);  
7.     printf("\n ...:%f",cv);  
8.     float dt = DienTich(tg);  
9.     printf("\n ...:%f",dt);  
10.    DIEM G=TrongTam(tg);  
11.    printf("\n...:");  
12.    Xuat(G);  
13. }
```

HƯỚNG DẪN BÀI 7

♦ Định nghĩa hàm

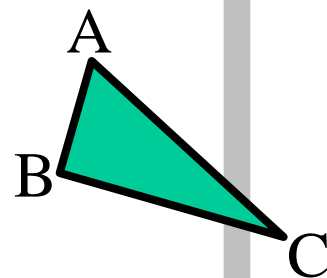
```
1. void Nhap (DIEM &P)
2. {
3.     float temp;
4.     printf("Nhap x: ");
5.     scanf("%f", &temp);
6.     P.x = temp;
7.     printf("Nhap y: ");
8.     scanf("%f", &temp);
9.     P.y = temp;
10. }
```



HƯỚNG DẪN BÀI 7

♦ Định nghĩa hàm

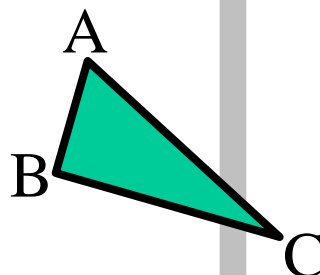
```
1. void Nhap(TAMGIAC &t)
2. {
3.     printf("Nhap A: ");
4.     Nhap(t.A);
5.     printf("Nhap B: ");
6.     Nhap(t.B);
7.     printf("Nhap C: ");
8.     Nhap(t.C);
9. }
```



HƯỚNG DẪN BÀI 7

♦ Định nghĩa hàm

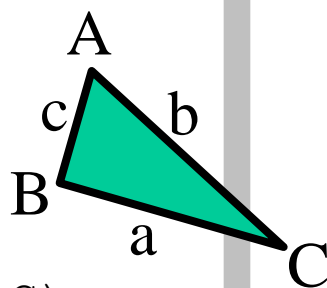
```
1. void Xuat (DIEM P)
2. {
3.     printf("\n x=%f", P.x);
4.     printf("\n y=%f", P.y);
5. }
6. void Xuat (TAMGIAC t)
7. {
8.     printf("\n A: ");
9.     Xuat (t.A);
10.    printf("\n B: ");
11.    Xuat (t.B);
12.    printf("\n C: ");
13.    Xuat (t.C);
14. }
```



HƯỚNG DẪN BÀI 7

♦ Định nghĩa hàm

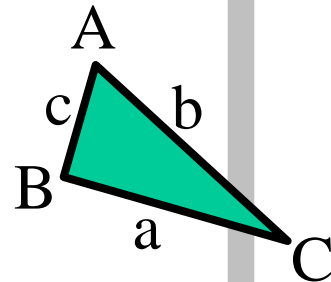
```
1. float KhoangCach(DIEM P, DIEM Q)
2. {
3.     return sqrt((P.x-Q.x)*(P.x-Q.x)
4.                 + (P.y-Q.y)*(P.y-Q.y));
5. }
6. float ChuVi(TAMGIAC t)
7. {
8.     float a = KhoangCach(t.B, t.C);
9.     float b = KhoangCach(t.C, t.A);
10.    float c = KhoangCach(t.A, t.B);
11.    return (a+b+c);
12. }
```



HƯỚNG DẪN BÀI 7

♦ Định nghĩa hàm

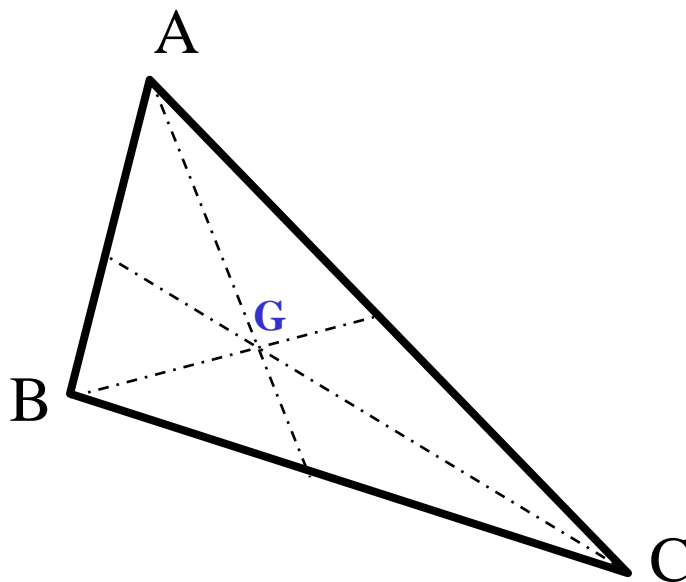
```
1. float KhoangCach (DIEM P, DIEM Q)
2. {
3.     return sqrt((P.x-Q.x)*(P.x-Q.x)
4.                 + (P.y-Q.y)*(P.y-Q.y));
5. }
6. float DienTich (TAMGIAC t)
7. {
8.     float a = KhoangCach (t.B, t.C);
9.     float b = KhoangCach (t.C, t.A);
10.    float c = KhoangCach (t.A, t.B);
11.    float p = (a+b+c) / 2;
12.    return sqrt(p*(p-a)*(p-b)*(p-c));
13. }
```



HƯỚNG DẪN BÀI 7

♦ Định nghĩa hàm

```
1. DIEM TrongTam (TAMGIAC t)
2. {
3.     DIEM temp;
4.     temp.x = (t.A.x+t.B.x+t.C.x)/3;
5.     temp.y = (t.A.y+t.B.y+t.C.y)/3;
6.     return temp;
7. }
```



HƯỚNG DẪN BÀI 8

- ♦ **Bài 8: Viết chương trình nhập tọa tâm và bán kính của một đường tròn. Tính diện tích và chu vi của đường tròn.**

```
1. #include <stdio.h>
2. #include <conio.h>
3. struct diem
4. {
5.     float x;
6.     float y;
7. };
8. typedef struct diem DIEM;
```


HƯỚNG DẪN BÀI 8

```
1. struct duongtron
2. {
3.     |    DIEM I;
4.     |    float R;
5. };
6. typedef struct duongtron DUONGTRON;
7. void Nhap(DIEM&) ;
8. void Xuat(DIEM) ;
9. void Nhap(DUONGTRON&) ;
10. void Xuat(DUONGTRON) ;
11. float ChuVi(DUONGTRON) ;
12. float DienTich(DUONGTRON) ;
```

HƯỚNG DẪN BÀI 8

```
1. void main()  
2. {  
3.     DUONGTRON c;  
4.     Nhap(c);  
5.     Xuat(c);  
6.     float cv = ChuVi(c);  
7.     printf("\n ...:%f",cv);  
8.     float dt = DienTich(c);  
9.     printf("\n ...:%f",dt);  
10. }
```

HƯỚNG DẪN BÀI 8

◆ Định nghĩa hàm

```
1. void Nhap (DIEM &P)
2. {
3.     float temp;
4.     printf("Nhap x: ");
5.     scanf("%f", &temp);
6.     P.x = temp;
7.     printf("Nhap y: ");
8.     scanf("%f", &temp);
9.     P.y = temp;
10. }
```

HƯỚNG DẪN BÀI 8

♦ Định nghĩa hàm

```
1. void Nhap (DUONGTRON &c)
2. {
3.     float temp;
4.     printf("Nhap tam: ");
5.     Nhap (c.I) ;
6.     printf("Nhap ban kinh: ");
7.     scanf ("%f", &temp) ;
8.     c.R = temp;
9. }
```

HƯỚNG DẪN BÀI 8

♦ Định nghĩa hàm

```
1. void Xuat(DIEM P)
2. {
3.     printf("\n x=%f", P.x);
4.     printf("\n y=%f", P.y);
5. }

6. void Xuat(DUONGTRON c)
7. {
8.     printf("\n Tam: ");
9.     Xuat(c.I);
10.    printf("Ban kinh:%f", c.R);
11. }
```

HƯỚNG DẪN BÀI 8

♦ Định nghĩa hàm

```
1. float ChuVi (DUONGTRON c)
2. {
3.     |    return 2*3.14*c.R;
4. }
```

HƯỚNG DẪN BÀI 8

♦ Định nghĩa hàm

```
1. float DienTich(DUONGTRON c)
2. {
3.     |    return 3.14*c.R*c.R;
4. }
```

HƯỚNG DẪN BÀI 9

- ♦ **Bài 9: Viết chương trình nhập vào hai phân số. Tính tổng, hiệu, tích, thương giữa chúng và xuất kết quả.**

```
1. #include <stdio.h>
2. #include <conio.h>
3. struct phanso
4. {
5.     int tu;
6.     int mau;
7. };
8. typedef struct phanso PHANSO;
```


HƯỚNG DẪN BÀI 9

1. void Nhap (PHANSO&) ;
2. void Xuat (PHANSO) ;
3. PHANSO Tong (PHANSO, PHANSO) ;
4. PHANSO Hieu (PHANSO, PHANSO) ;
5. PHANSO Tich (PHANSO, PHANSO) ;
6. PHANSO Thuong (PHANSO, PHANSO) ;

HƯỚNG DẪN BÀI 9

```
1. void main()  
2. {  
3.     PHANSO a,b,kq;  
4.     printf("\n Nhap...:");  
5.     Nhap(a);  
6.     printf("\n Nhap...:");  
7.     Nhap(b);  
8.     kq = Tong(a,b);  
9.     printf("\n Tong...:");  
10.    Xuat(kq);  
11.    kq = Hieu(a,b);  
12.    printf("\n Hieu...:");  
13.    Xuat(kq);  
14.    kq = Tich(a,b);  
15.    printf("\n Tich...:");  
16.    Xuat(kq);  
17. }
```

HƯỚNG DẪN BÀI 9

♦ Định nghĩa hàm nhập phân số

```
1. void Nhap(PHANSO &x)
2. {
3.     printf("Nhap tu: ");
4.     scanf("%d",&x.tu);
5.     printf("Nhap mau: ");
6.     scanf("%d",&x.mau);
7. }
```

HƯỚNG DẪN BÀI 9

♦ Định nghĩa hàm xuất phân số

```
1. void Xuat(PHANSO x)
2. {
3.     printf("\n Tu: %d", x.tu);
4.     printf("\n Mau: %d", x.mau);
5. }
```

HƯỚNG DẪN BÀI 9

$$\begin{array}{c} \textcolor{blue}{x} \\ \textcircled{\frac{a}{b}} \end{array} + \begin{array}{c} \textcolor{blue}{y} \\ \textcircled{\frac{c}{d}} \end{array} = \begin{array}{c} \textcolor{blue}{temp} \\ \textcircled{\frac{ad+bc}{bd}} \end{array}$$

◆ Định nghĩa hàm

```

1. PHANSO Tong (PHANSO x, PHANSO y)
2. {
3.     PHANSO temp;
4.     temp.tu=x.tu*y.mau + x.mau*y.tu;
5.     temp.mau=x.mau*y.mau;
6.     return temp;
7. }
```

HƯỚNG DẪN BÀI 9

$$\begin{array}{c} \textcolor{blue}{x} \\ \textcircled{\frac{a}{b}} \end{array} - \begin{array}{c} \textcolor{blue}{y} \\ \textcircled{\frac{c}{d}} \end{array} = \begin{array}{c} \textcolor{blue}{temp} \\ \textcircled{\frac{ad-bc}{bd}} \end{array}$$

◆ Định nghĩa hàm

```

1. PHANSO Hieu(PHANSO x, PHANSO y)
2. {
3.     PHANSO temp;
4.     temp.tu = x.tu*y.mau-x.mau*y.tu;
5.     temp.mau = x.mau*y.mau
6.     return temp;
7. }
```

HƯỚNG DẪN BÀI 9

$$\begin{array}{c} \textcolor{blue}{x} \\ \textcircled{\frac{a}{b}} \end{array} * \begin{array}{c} \textcolor{blue}{y} \\ \textcircled{\frac{c}{d}} \end{array} = \begin{array}{c} \textcolor{blue}{temp} \\ \textcircled{\frac{ac}{bd}} \end{array}$$

◆ Định nghĩa hàm

```
1. PHANSO Tich (PHANSO x, PHANSO y)
2. {
3.     PHANSO temp;
4.     temp.tu=x.tu*y.tu;
5.     temp.mau=x.mau*y.mau
6.     return temp;
7. }
```

HƯỚNG DẪN BÀI 9

$$\begin{array}{c} \textcolor{blue}{x} \\ \textcircled{\frac{a}{b}} \end{array} / \begin{array}{c} \textcolor{blue}{y} \\ \textcircled{\frac{c}{d}} \end{array} = \begin{array}{c} \textcolor{blue}{temp} \\ \textcircled{\frac{ad}{bc}} \end{array}$$

◆ Định nghĩa hàm

```
1. PHANSO Thuong (PHANSO x, PHANSO y)
2. {
3.     PHANSO temp;
4.     temp.tu = x.tu*y.mau;
5.     temp.mau = x.mau*y.tu;
6.     return temp;
7. }
```